

**In the Specification:**

At page 4, lines 4-14, please amend the paragraph as follows:

In another specific embodiment of the invention involving a flow-identification arrangement that is susceptible to unbalanced bandwidth allocation, queue management is realized using a stateless FIFO queue and a server including a CPU arrangement. The FIFO queue is configured and arranged to receive packets having have associated flow identification information. The CPU arrangement in the server is adapted to detect a matching flow identification between a recently-received incoming packet with at least one packet selected from a set of outgoing packets, the packet being selected as a function of a random probability. In response to the matching flow identification detection, the CPU arrangement is further adapted to mitigate unbalanced bandwidth allocation without maintaining state information for the FIFO queue, by reducing the processing priority of said at least one selected packet and the recently-received packet.

At pages 4-5, lines 15-31 and 1-3 respectively, please amend the paragraph at line 16 as follows:

Another more specific embodiment of the present invention is directed to bridging fairness and simplicity in active queue management. In this specific embodiment, Specifically, active queue management algorithm is realized by differentially penalizing misbehaving flows by dropping more of their packets and without requiring management of state information for the FIFO buffer through which the packets flow. In this manner, by keeping packets of responsive flows and dropping packets of unresponsive flows, the embodiment aims to approximate max-min fairness for the flows that pass through a congested router. Using the FIFO buffer, the contents of the FIFO buffer are used to form a “sufficient statistic” about the incoming traffic and this statistic is used in a simple fashion to penalize misbehaving flows. When a packet arrives at a congested router, a packet is drawn randomly from the FIFO buffer and it is compared with the arriving packet. If the randomly-drawn packet and the arriving packet belong to the same flow, then they are both dropped. If the randomly-drawn packet and the arriving packet do not belong to the same flow, the randomly chosen packet is left intact and the arriving packet is admitted into the buffer with a probability that depends on the level of congestion. In this context, the allocation of the flows which consume the most resources are identified and reduced, thereby attempting to minimize the resource consumption of the maximum flow so as to achieve the degree of “minmax fairness” recognized by the previous approaches. The resource freed up as a result of minimizing the maximum flow’s consumption is distributed among the other flows. In the Internet context, the former flows are either unfriendly TCP or UDP, and the latter flows are TCP.

At pages 7-8, lines 30-31 and 1-16 respectively, please amend the paragraph at line 7 by inserting the mathematical symbol  $\geq$  between  $m$  and  $1$  as follows

It has also been discovered in connection with the present invention that, as the number of unresponsive flows increases, it is effective to choose to drop more candidate packets. Accordingly, in connection with another embodiment of the present invention, the above-described process flow is modified in that a selected increased number of candidate packets are dropped in a manner that corresponds to an increased detected number of unresponsive flows. When implementing an embodiment of the present invention as a stateless design, there is not priori knowledge of how many unresponsive flows are active at any time for choosing a suitable value for  $m$ . However, yet another aspect of the present invention can be implemented to permit the process to be automated so that the algorithm chooses the proper value of  $m \geq 1$ . One way of achieving this is to introduce an intermediate threshold  $int_{th}$  which partitions the interval between the  $min_{th}$  and  $max_{th}$  into two regions. For example, when the average buffer occupancy is between the  $min_{th}$  and  $int_{th}$ , the algorithm sets  $m = 1$  and when the average buffer occupancy is between the  $int_{th}$  and  $max_{th}$  the algorithm sets  $m = 2$ . When the buffer occupancy exceeds  $max_{th}$ ,  $m$  remains the same but each incoming packet is dropped. More generally, multiple thresholds can be introduced to partition the interval between the  $min_{th}$  and  $max_{th}$  into  $k$  regions  $R_1, R_2, \dots, R_k$  and choose different values of  $m$  depending on the region of the average buffer occupancy falls in. For example, choose  $m = 2 \cdot i$  ( $i = 1, \dots, k$ ), when the average queue size lies in region  $R_i$ , and let  $m$  increase monotonically with the average queue size.

At pages 8-9, lines 29-31 and 1-19 respectively, please amend the paragraph as follows:

As mentioned above, the present invention is applicable to a spectrum of network configurations and traffic mixes, and to a variety of application types that route identification-bearing data sets (or “packets”) in a communication environment. These communication environments include but are not necessarily limited to single and multiple congested links and single and multiple misbehaving flows. FIG. 2 illustrates one such environment in which a network 200 is configured with one or both routers 206 and 208 (*R*1 and *R*2) having a CPU programmed with process flow 100 to choose and drop packets as described in connection with one of the above queue management implementations. The network 200 has a single link 210, between the routers 206 and 208, that is susceptible to undesirable levels of congestion. In this example environment, the link 210 has a capacity of one Mbps and is shared by *m* TCP and *n* UDP flows, depicted by the corresponding sources 212, 214 and corresponding sinks 216 and 218. An end host (not illustrated) is connected to the routers using a ten-Mbps link, which is ten times the bottleneck link bandwidth. All links have a small propagation delay of one millisecond so that the delay experienced by a packet is mainly caused by the buffer delay rather than the transmission delay. The maximum window size of TCP is set to three hundred such that it does not become a limiting factor of a flow’s throughput. The TCP flows are derived from FTP sessions that transmit large sized files. The UDP hosts send packets at a constant bit rate (CBR) of *r* Kbps, where *r* is a positive integer variable. All packets are set to have a size of 1K Bytes. For a discussion of performance and other related application-specific details, reference may be made to the article ~~appended hereto~~ and entitled, “CHOKe: A stateless active queue management scheme for approximating fair bandwidth allocation,” as was filed in the above-referenced Provisional Patent Document, Serial No. 60/185,569 filed on February 28, 2000.

At page 9, lines 20-27, please amend the paragraph as follows:

While some of the above-discussed embodiments involve choosing the drop candidate packet from the queue in a manner that is based on a random probability, alternative embodiments involve choosing the drop candidate packet in other manners. For example, the drop candidate packet can be chosen always as the packet at the head of the queue, and in yet another embodiment, chosen always as the packet at the tail of the queue. These last two variations are reasonable approximations of the embodiments which choose the drop candidate packet from the queue randomly. The above-referenced article in the underlying provisional patent document appended hereto further discusses these last two variations.